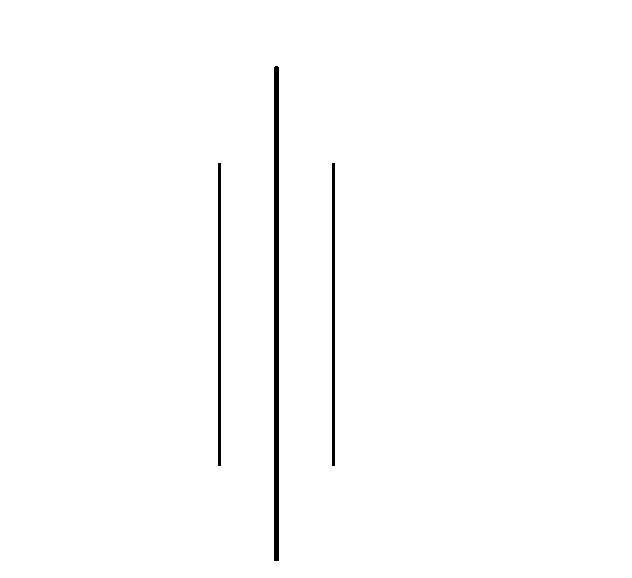
NEPAL ENGINEERING COLLEGE

( Affiliated To Pokhara University )

Changunarayan, Bhaktapur



Report on

# Lab 3: Filters in Images

SUBMITTED BY : SUBMITTED TO:

Name: Rishi Raj Pandey Department of Computer Roll no: 021 – 366 Science and Engineering Date: 11 / 02 / 2082 Department of Computer Science and Engineering

**Objectives**: To perform Mean or Low Pass Filter, High Pass Filter, Gaussian Filter and Laplacian Filter on an RGB image and save the resulting images in various file formats.

**Convert the Given RGB image into**

* Load Image
* Mean or Low Pass Filter
* High Pass Filter
* Gaussian Filter
* Laplacian Filter
* Save image in different formats

**Code**

namespace Lab3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

pictureBox2.SizeMode = PictureBoxSizeMode.StretchImage;

}

Bitmap OriginalPic;

private void Load\_Click(object sender, EventArgs e)

{

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == DialogResult.OK)

{

pictureBox1.Image = new Bitmap(ofd.FileName);

OriginalPic = ToGreyScale((Bitmap)pictureBox1.Image);

ShowImage(OriginalPic, "Original Image");

}

}

static Bitmap ToGreyScale(Bitmap original)

{

Bitmap grey = new Bitmap(original.Width, original.Height);

for (int x = 0; x < original.Width; x++)

{

for (int y = 0; y < original.Height; y++)

{

Color pixel = original.GetPixel(x, y);

int greyVal = (int)(0.3 \* pixel.R + 0.59 \* pixel.G + 0.11 \* pixel.B);

grey.SetPixel(x, y, Color.FromArgb(greyVal, greyVal, greyVal));

}

}

return grey;

}

static void ShowImage(Bitmap bmp, string title)

{

Form form = new Form();

form.Text = title;

form.ClientSize = new Size(bmp.Width, bmp.Height);

PictureBox pb = new PictureBox();

pb.Dock = DockStyle.Fill;

pb.Image = bmp;

pb.SizeMode = PictureBoxSizeMode.StretchImage;

form.Controls.Add(pb);

form.Show();

}

private Bitmap MeanFilterr(Bitmap bmp)

{

Bitmap result = new Bitmap(bmp.Width, bmp.Height);

for (int y = 1; y < bmp.Height - 1; y++)

{

for (int x = 1; x < bmp.Width - 1; x++)

{

int sumR = 0, sumG = 0, sumB = 0;

for (int j = -1; j <= 1; j++)

{

for (int i = -1; i <= 1; i++)

{

Color pixel = bmp.GetPixel(x + i, y + j);

sumB += pixel.B;

sumG += pixel.G;

sumR += pixel.R;

}

}

int r = sumR / 9;

int g = sumG / 9;

int b = sumB / 9;

result.SetPixel(x, y, Color.FromArgb(r, g, b));

}

}

return result;

}

private void MeanFilter\_Click\_2(object sender, EventArgs e)

{

if (pictureBox1.Image != null)

{

pictureBox2.Image = MeanFilterr(OriginalPic);

ShowImage((Bitmap)pictureBox2.Image, "Using Mean Filter)");

}

}

private Bitmap HighPassFilter(Bitmap bmp)

{

Bitmap result = new Bitmap(bmp.Width, bmp.Height);

int[,] kernel = {

{ -1, -1, -1},

{ -1, 8, -1},

{ -1, -1, -1}

};

for (int y = 1; y < bmp.Height-1; y++)

{

for (int x = 1; x < bmp.Width-1; x++)

{

int sumR = 0, sumG = 0, sumB = 0;

for (int j = -1; j <= 1; j++)

{

for (int i = -1; i <= 1; i++)

{

Color pixel = bmp.GetPixel(x + i, y + j);

int k = kernel[j + 1, i + 1];

sumR += pixel.R\*k;

sumG += pixel.G\*k;

sumB += pixel.B\*k;

}

}

sumR = Math.Min(Math.Max(sumR, 0), 255);

sumG = Math.Min(Math.Max(sumG, 0), 255);

sumB = Math.Min(Math.Max(sumB, 0), 255);

result.SetPixel(x, y, Color.FromArgb(sumR, sumG, sumB));

}

}

return result;

}

private Bitmap ApplyLaplacianFilter(Bitmap gray)

{

int[,] kernel = {

{ 0, -1, 0},

{ -1, 4, -1},

{ 0, -1, 0}

};

int width = gray.Width;

int height = gray.Height;

Bitmap result = new Bitmap(gray.Width, gray.Height);

for (int x = 1; x < width-1; x++)

{

for (int y = 1; y < height-1; y++)

{

int sum = 0;

for (int i = -1; i <= 1; i++)

{

for (int j = -1; j <= 1; j++)

{

int pixelVal = gray.GetPixel(x + i, y + j).R;

sum += pixelVal \* kernel[i + 1, j + 1];

}

}

sum = Math.Min(255, Math.Max(0, sum));

Color edgeColor = Color.FromArgb(sum, sum, sum);

result.SetPixel(x, y, edgeColor);

}

}

return result;

}

private void button3\_Click(object sender, EventArgs e)

{

if (OriginalPic != null)

{

pictureBox2.Image = HighPassFilter(OriginalPic);

ShowImage((Bitmap)pictureBox2.Image, "High Pass Filter");

}

}

private void button5\_Click(object sender, EventArgs e)

{

if (OriginalPic != null)

{

pictureBox2.Image = ApplyLaplacianFilter(OriginalPic);

ShowImage((Bitmap)pictureBox2.Image, "Laplacian Filter");

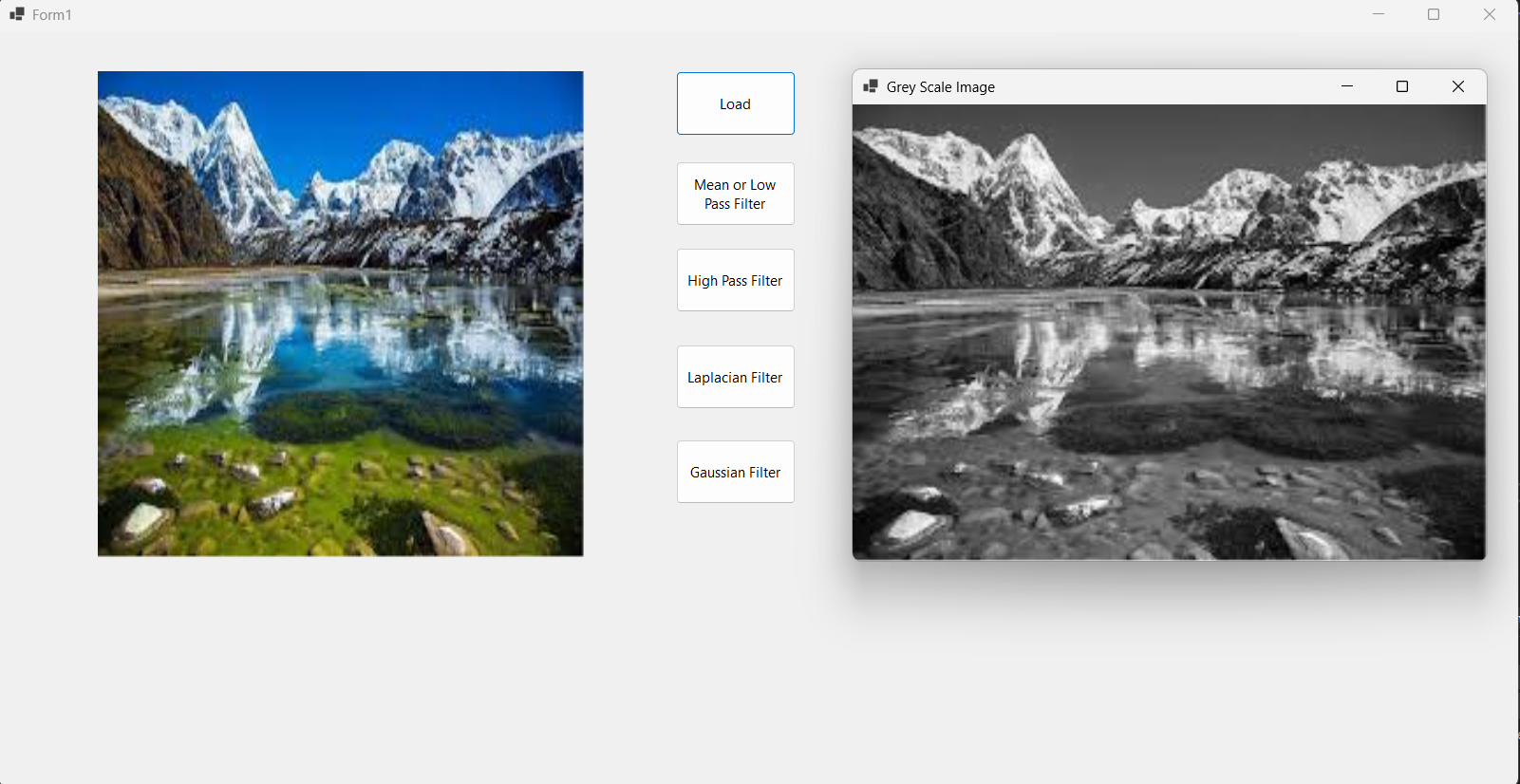
}

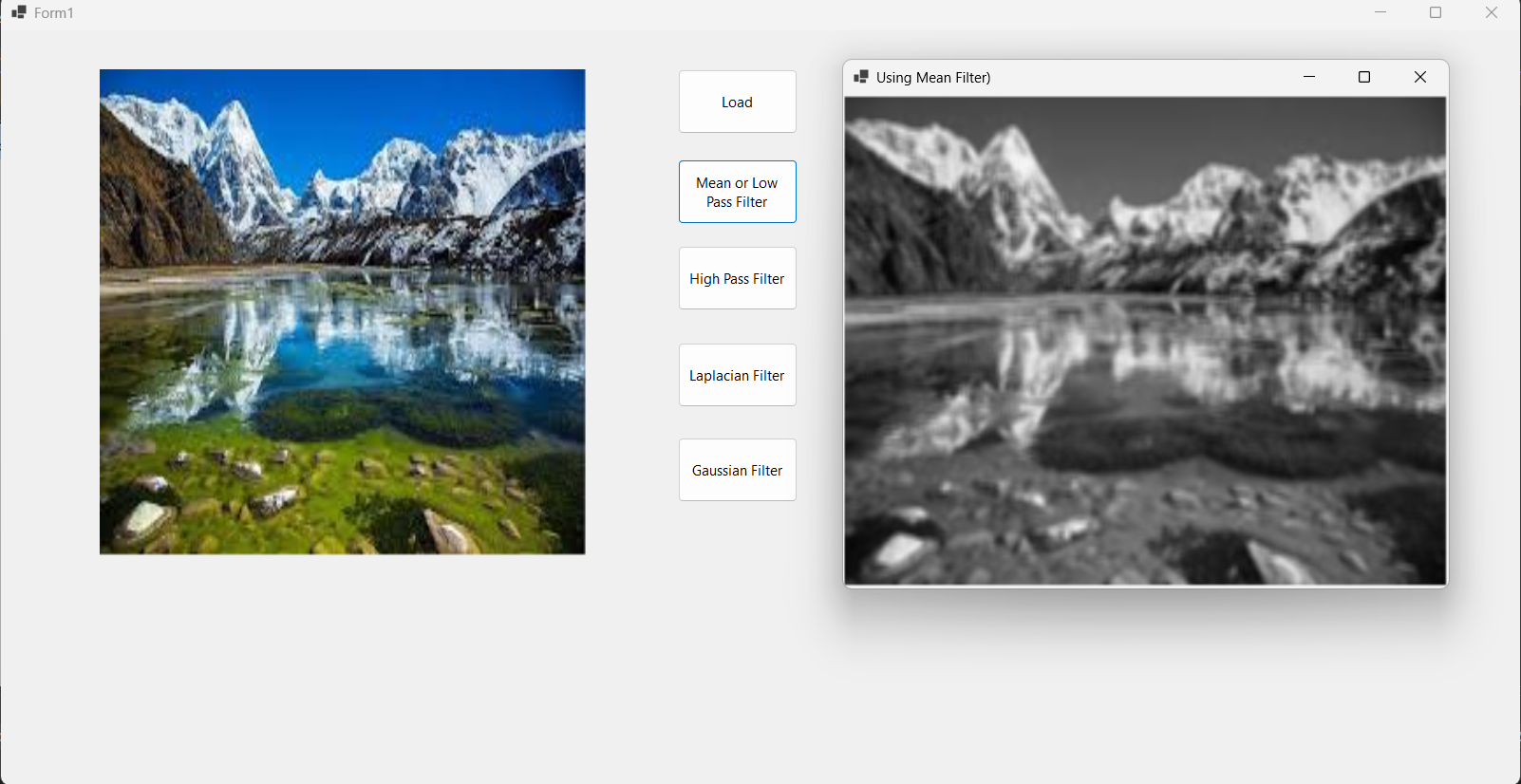
}

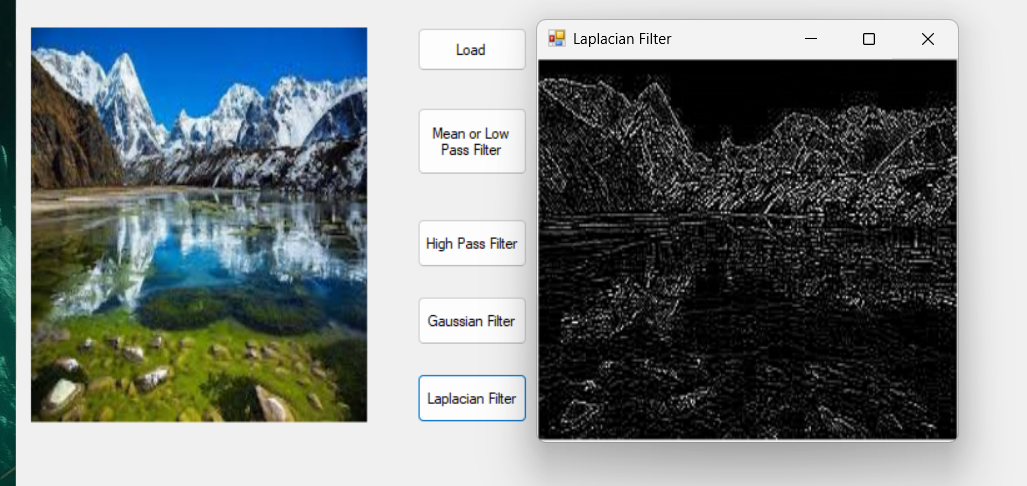
}

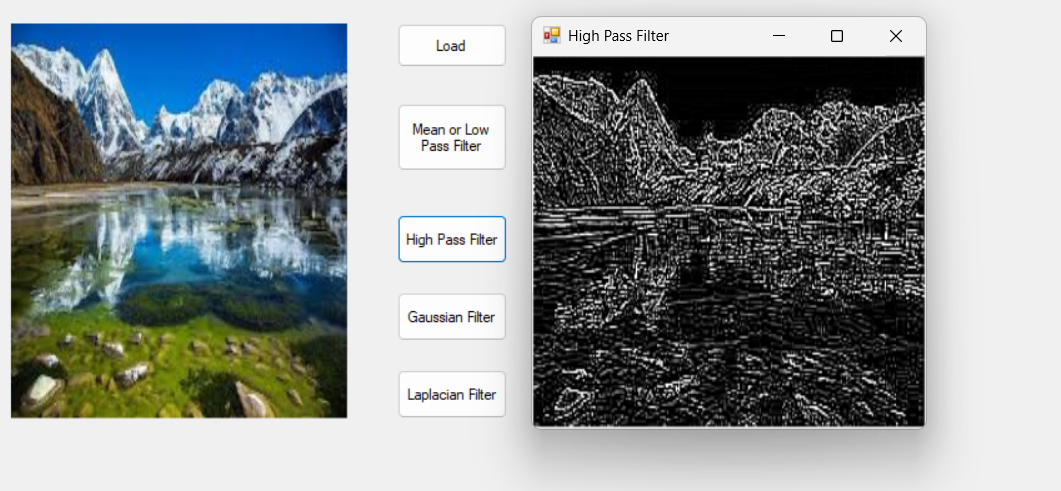
}

**OUTPUT**

****

****

****

****

**OBSERVATION**

The RGB image was successfully loaded using libraries like OpenCV or PIL, preserving its color channels and serving as the input for filtering operations.

Applying the **Mean (Low Pass) Filter** resulted in a smoother image with reduced noise and softened details, indicating suppression of high-frequency content.

The **High Pass Filter** enhanced edges and textures, making the image appear sharper while suppressing smooth regions.

The **Laplacian Filter** highlighted edges and transitions by emphasizing regions of rapid intensity change, enhancing contours and outlines.

All filtered images were saved in multiple formats (PNG, JPEG, BMP), with minor differences in quality and file size based on the format.

**CONCLUSION**

The experiment demonstrated the effective use of various filters on an RGB image. Mean filters smoothed the image, while High Pass and Laplacian filters enhanced edges. Gaussian filtering offered the best trade-off between smoothing and detail retention.

Saving the images in different formats showed how compression affects image quality and size. Overall, this task provided practical insight into filtering techniques used in noise reduction, edge detection, and image enhancement.